



TITLE:

有効数係数多項式(数式処理における理論とその応用の研究)

AUTHOR(S):

鈴木, 正幸

CITATION:

鈴木, 正幸. 有効数係数多項式(数式処理における理論とその応用の研究). 数理解析研究所講究録 1996, 941: 175-184

ISSUE DATE:

1996-03

URL:

<http://hdl.handle.net/2433/60122>

RIGHT:

22.

有効数係数多項式

鈴木正幸
(岩手大学・工学部)

22.1 はじめに

本稿では、近似係数の多項式算法を確実に計算することを目指す。これまでに、浮動小数係数多項式の近似算法の解析 [1], 区間数を用いた Gröbner 基底の計算 [2], 近似算法の安定化理論の試み [3][4] などの研究がある。

筆者らは、精度がある桁まで保証されている係数を持つ多項式を入力とし、精度の保証された計算と、精度に関する構成の研究を行ってきた。べき級数係数の場合は [5][6] に示した。今回は浮動少数係数多項式を扱う。

[5][6] を簡単にまとめておく。通常モジュラ算法では、多項式 $F \in K[y, \dots, z][x]$ の計算を $S = (y, \dots, z)$ と法を選び、 $(\text{mod } S)$ で計算した $F^{(0)}$ を出発多項式として、 $F^{(i)} \equiv F(\text{mod } S^i)$, $i = 1, \dots$ を構成していく。ここで現われる $F^{(i)}$ を F の近似と考えることができる。

モジュラ算法において、 F の主係数に定数項がなければ、このような近似と構成法は適用できない。構成の出発多項式として、 $F^{(k)}$ から始めると F を構成することができるが、 $F^{(k)}$ を次の計算に用いる場合、同じ法を用いて計算することができなくなる。この問題の原因は、多項式全体に同じ法を取ることで、また、計算全体を同じ法で行なうことである。

[5][6] では、有効精度という考えを示し、係数毎に有効な法を定めた多項式を提案した。有効精度に基づく計算方法、計算による各係数の法の変化、正しい近似計算を行なうための方法、そして近似を上げていく構成法を示した。

本稿では、この理論に基づき、精度として有効桁を用いた場合について考察する。精度がある桁ま

で保証されている係数を持つ多項式を入力とし、精度の保証された計算と、精度に関する構成を行なうのが目的である。筆者の知るかぎり、このような数係数を扱える数式処理システムは存在しない。区間数を扱えるシステムが最も近いものであるが、多倍長であること、構成のため結果の精度が保証できる所まで打ち切ること、が必要となる。そのため、有効桁による精度保証が行なえる数システムと、それを係数とする多項式処理システムを作成した。22.2で有効数演算を定義し、22.3で剰余列算法例を示す。

22.2 有効数と演算

近似係数を扱う場合、数のデータ型によらず、正確な表現なのかどうか重要である。数を正確に表現された数(正確数)と、正確には表現されていない数(不正確数)に区別し、不正確数に対し、次のことを要請する:

- 計算誤差の混入を防ぐこと,
- 演算結果の精度保証を行なうこと,
- 計算精度の制御ができること,
- 精度に関する構成ができること.

これらの要求に対し、不正確数として次の有効数を用いることを提案する。

22.2.1 有効数

有効数は次の三組で表現される: $(m, N, e) = N \times B^e$. ここで, N は正確数, m は N の有効桁, e は打ち切り次数, B は基数(radix)である. これを \tilde{N}_m^e と表記し, 真の値 N に対し,

$$0 \leq N - \tilde{N}_m^e \leq B^e$$

と解釈する. 例えば, B を 10 とし, 123.4500 が 10^{-4} で打ち切られたとすると, $1234500\bar{7}^{-4}$ と表し, $123.4500 \leq 1234500\bar{7}^{-4} \leq 123.4501$ である.

有効数に対する演算の結果, 有効桁がなくなった(有効数 \tilde{N}_m^e の N が 0 となった)場合は, 結果を有効ゼロとする. 有効ゼロを $\tilde{0}^e$ と表し,

$$-B^e \leq \tilde{0}^e \leq B^e$$

と解釈する.

22.2.2 有効演算

有効数に対する演算を有効演算と呼び, 加算を有効加算, 等々と呼ぶことにする. 有効演算には演算記号の下に \sim をつけて \pm, \sim, \sim, \sim と表す. 以下では見やすさのために, 有効桁数は省くことに

する。通常の数をも B^e で切り捨て、正規化し、有効数とする演算を $\lceil \cdot \rceil^e$ と表す。このとき、有効桁がない場合は、有効ゼロ $\tilde{0}^e$ を結果とする。

有効演算を、有効ゼロの場合を含めて、次のように定義する：

$$\begin{aligned}\tilde{N}^{e_n} \pm \tilde{M}^{e_m} &= \lceil \tilde{N}^{e_n} \pm \tilde{M}^{e_m} \rceil^e, & e = \max(e_n, e_m) + 1 \\ \tilde{N}^{e_n} \times \tilde{M}^{e_m} &= \lceil \tilde{N}^{e_n} \times \tilde{M}^{e_m} \rceil^e, & e = e_n + e_m + 1 \\ \tilde{N}^{e_n} \oslash \tilde{M}^{e_m} &= \begin{cases} Undefined, & \tilde{M}^{e_m} \text{が有効ゼロ,} \\ \lceil \tilde{N}^{e_n} / (\tilde{M}^{e_m} \pm \tilde{B}^{e_m}) \rceil^e, & e = e_n - e_m + 1. \end{cases}\end{aligned}$$

つまり、怪しい桁以降は切り捨て演算結果の有効桁を保証し、演算結果が精度落ちし、有効桁が無くなった場合、どの次数まで 0 かを保証する。

22.2.3 有効数係数多項式

係数が正確数あるいは有効数である多項式を有効数係数多項式と呼ぶ。通常が多項式と異なるのは、主係数が有効ゼロになる場合である。正確に 0 になる場合を除き、有効ゼロを係数に持つ項は計算結果に残すことにする。したがって、多項式算法が次数に関する演算や除算を含む場合は注意を要する。有効係数多項式を 0 と判定するのは、係数の全てが 0 または有効ゼロになった場合とする。

22.2.4 有効数係数に関する構成

計算に誤差が入らないので、演算の有効桁に関する構成が可能となる。[6] において、計算グラフによる精度に関する構成法を示した。ここでも同様に、

計算入力

無限桁まで保証されている有効数係数の多項式

初期計算 (計算グラフの生成)

ある精度で入力を切りだし、有効演算を行なう。この時、演算の入出力に関する計算グラフを作成する。入出力有効計算値、演算を保持し、後の構成に利用する。

有効桁に関する演算の構成 (計算グラフによる構成)

結果が望ましい精度で得られなかった場合、その原因は二つある。(1) 演算における桁落ち、(2) 除算不能 (徐多項式の主係数が有効ゼロ) の場合である。

加減乗算については有効計算は各係数毎の有効桁について、正しい近似計算になっている。従って、入力の有効桁を 1 上げて、高精度の項のみを計算すれば、無駄な計算なしに結果の有効桁を 1 上げることができる。除算も除算可能であれば、同様である。

有効除算が可能でない場合は、(1) 徐多項式の主係数についてのみ有効桁を上げていくか、(2) 主

係数をゼロとみなすか、を選べる。選択は算法による。

計算グラフ上から除算不能ノードをなくせば、結果から計算グラフを逆にたどり、有効桁を1ずつ上げて行けばよい。

22.3 有効数係数多項式の剰余列計算

有効数係数多項式計算システム

既存の数式処理システム上に有効数システムを実現するのが難しかったため、Scheme 上に有効数と一変数多項式演算システムを作成した。総称演算機能を取り入れ、簡単に新しい係数体を追加できるように考慮した。現在、整数、有理数、有効数、区間数、一変数多項式が扱える。

人にとってのわかりやすさ、算法の正しさの確認の容易性から基数を 10 にとっている。しかしそのため、計算の有効桁落ちが大きい点は問題である。

有効剰余列算法

正確に与えられた P_1 と P_2 をある有効桁 (a) で切り出して、有効数係数多項式とし、通常の剰余列算法で剰余列を求める。この剰余列のことを有効剰余列と呼び、求められた剰余列の最後の要素 P_i を有効 gcd と呼ぶ。以下に、本システムの有効剰余計算プログラム (Scheme) を示す。

剰余列の計算

```
(define (poly:prs p1 p2)
  (let ((p3 0) (q 0))
    (while (and (not (<zero?> p2))
                (> (poly:deg p2) 0))
      (set! p3 (poly:prem p1 p2))
      (while (<zero?> (poly:lc p3))
        (set! p3 (poly:rest p3)))
      (set! p1 p2)
      (set! p2 p3))
    (cond ((<zero?> p2) p1)
          (else p2))))
```

剰余の計算

```
(define (poly:prem p1 p2)
  (let ((mv (poly:mvar p1)) (lc (poly:lc p2))
        (d1 (poly:deg p1)) (d2 (poly:deg p2))
        (qc 0) (xt 0))
    (while (>= d1 d2)
      (set! qc (</> (poly:lc p1) lc))
      (set! xt (poly:make mv '((- d1 d2) . .qc))))
    (set! p1 (<-> (poly:rest p1)
                  (<*> (poly:rest p2) xt)))
    (set! d1 (poly:deg p1)))
  p1))
```

<> で囲まれている関数は総称演算を表す。例えば、<zero?> はゼロ判定を行なう総称関数で、0、有効ゼロ、係数が全て<zero?>である多項式、のときに真を返す。また、poly: が前に付いている関数は多項式に関する演算を表す。ここで示したプログラムは、通常の剰余列プログラムとほとんど同じである。

22.3.1 有効剰余列計算

以下に、近似根や重根を持つ場合の計算例 (ex1 ~ ex5) を示すが、この計算と出力は、

- 有効桁を1つつ上げながら、それぞれの有効桁での有効剰余列を求めた。
- 計算の有効桁の違いにより、異なる次数の有効 gcd を求めた。
- 同一次数で最大有効桁の有効 gcd を表示。括弧内は出発桁を表す。

例中、有効数表記 \tilde{N}_m^e の代わりに、 \tilde{N}_m^e の浮動少数表記を f として、 \tilde{f}_m^{ee} と表記してある。

■ ex.1: 近接根

ex.1-(a)

$$\begin{aligned} p_1 &= (x+1)(x-2)(x-1.5) \\ &\quad \times (x-0.500)(x-0.502) \\ p_2 &= (x-1)(x+2)(x+1.5) \\ &\quad \times (x-0.501)(x-0.503) \end{aligned}$$

ex.1-(b)

$$\begin{aligned} p_1 &= (x+1)(x-2)(x-1.5) \\ &\quad \times (x-0.501)(x-0.502) \\ p_2 &= (x-1)(x+2)(x+1.5) \\ &\quad \times (x-0.502)(x-0.503) \end{aligned}$$

ex.1-(a), ex.1-(b) ともに、 $x = 0.5$ の辺りに近接根を持つ。ex.1-(a) の p_1 と p_2 は (真の) gcd を持たないが、ex.1-(b) の p_1 と p_2 は真の gcd を持つ。両例ともある有効桁までは、 $(x-0.5)^2$ が有効 gcd として求まり、それ以上の有効桁では、ex.1-(a) は有効 gcd なし、ex.1-(b) は gcd として、 $(x-0.502)$ が求まることが期待される。

gcd.1-(a)

$$\begin{aligned} (7) \quad & 0.9999_4^{e-4} x^4 - 1.0031_5^{e-4} x^3 - 0.9486_4^{e-4} x^2 \\ & + 1.2039_5^{e-4} x + 0.3019_4^{e-4} \\ (11) \quad & 0.999_3^{e-3} x^3 - 1.004_4^{e-3} x^2 \\ & + 0.256_3^{e-3} x - 0.002_1^{e-3} \\ (17) \quad & 0.99999_5^{e-5} x^2 - 1.00163_6^{e-5} x + 0.25081_5^{e-5} \\ & \approx (x-0.5)^2 \\ (26) \quad & 0.99999999_8^{e-8} x - 0.50150156_8^{e-8} \\ & \approx (x-0.5) \\ (32) \quad & -4.49305e-6_6^{e-11} \end{aligned}$$

gcd.1-(b)

$$\begin{aligned} (7) \quad & 0.9999_4^{e-4} x^4 - 1.0041_5^{e-4} x^3 - 0.9481_4^{e-4} x^2 \\ & + 1.2051_5^{e-4} x - 0.3025_5^{e-4} \\ (11) \quad & 0.999_3^{e-3} x^3 - 1.005_4^{e-3} x^2 + 0.256_3^{e-3} x - 0.002_1^{e-3} \\ (17) \quad & 0.99999_5^{e-5} x^2 - 1.00263_6^{e-5} x + 0.25131_5^{e-5} \\ & \approx (x-0.5)^2 \\ (32) \quad & 0.99999999_14^{e-14} x - 0.50199999_14^{e-14} \\ & \approx (x-0.5) \end{aligned}$$

gcd.1-(a) では、有効桁 7 桁までが 4 次、8 ~ 11 桁までが 3 次、12 ~ 17 桁までが 2 次、18 ~ 26 桁までが 1 次、の有効 gcd が求められている。それぞれの有効桁で、次の次数の剰余は有効ゼロ多項式となっている。27 桁以降で有効 gcd は得られていない。gcd.1-(b) の例では、同様に、7 桁 p までが 4 次、8 ~ 11 桁までが 3 次、12 ~ 17 桁までが 2 次、18 ~ 26 桁までが 1 次、の有効 gcd が求め

られている。

有効 gcd が近似 gcd であることの判定法は次節で示し、ここでは直観的に行なう。両例とも有効桁 12 桁から、近似 gcd として $(x - 0.5)^2$ が得られる。有効桁 26 以上では、**ex.1-(a)** は近似 gcd なし、**ex.1-(b)** は $(x - 0.502)$ が近似 gcd として求まる。有効 gcd 中に、直観的な近似 gcd が含まれていることがわかる。

■ ex.2: より近い近接根

ex.2-(a)

$$\begin{aligned} p_1 &= (x+1)(x-2)(x-1.01) \\ &\quad \times (x-0.50000)(x-0.50002) \\ p_2 &= (x-1)(x+2)(x+1.01) \\ &\quad \times (x-0.50001)(x-0.50003) \end{aligned}$$

ex.2-(b)

$$\begin{aligned} p_1 &= (x+1)(x-2)(x-1.01) \\ &\quad \times (x-0.50000)(x-0.50002) \\ p_2 &= (x-1)(x+2)(x+1.01) \\ &\quad \times (x-0.50002)(x-0.50003) \end{aligned}$$

これらは、**ex.1** の近接根をもう少し近付けてみた例になっている。

gcd.2-(a)

$$\begin{aligned} (8) \quad & 0.999999^e {}_5^{-5} x^4 - 1.00002^e {}_6^{-5} x^3 \\ & - 0.75495^e {}_5^{-5} x^2 + 1.005^e {}_6^{-5} x + 0.25125^e {}_5^{-5} \\ (13) \quad & 0.9999^e {}_4^{-4} x^3 - 1.0006^e {}_5^{-4} x^2 \\ & + 0.2515^e {}_4^{-4} x - 4.0e - 4^e {}_1^{-4} \\ (19) \quad & 0.999999^e {}_6^{-6} x^2 - 0.999557^e {}_6^{-6} x + 0.249778^e {}_6^{-6} \\ & \approx (x-0.5)^2 \\ (32) \quad & 0.99999999^e {}_{12}^{-12} x - 0.50001500^e {}_{12}^{-12} \\ & \approx (x-0.5) \\ (36) \quad & -3.027e - 10^e {}_4^{-13} \end{aligned}$$

gcd.2-(b)

$$\begin{aligned} (8) \quad & 0.999999^e {}_5^{-5} x^4 - 1.00003^e {}_6^{-5} x^3 \\ & - 0.75495^e {}_5^{-5} x^2 + 1.00501^e {}_6^{-5} x - 0.25126^e {}_5^{-5} \\ (13) \quad & 0.9999^e {}_4^{-4} x^3 - 1.0006^e {}_5^{-4} x^2 \\ & + 0.2515^e {}_4^{-4} x - 4.0e - 4^e {}_1^{-4} \\ (19) \quad & 0.999999^e {}_6^{-6} x^2 - 0.999567^e {}_6^{-6} x + 0.249783^e {}_6^{-6} \\ & \approx (x-0.5)^2 \\ (36) \quad & 1.0^e {}_{16}^{-16} x - 0.50002^e {}_{16}^{-16} \\ & \approx (x-0.50002) \end{aligned}$$

ex.1 に比べて、同じ次数の有効 gcd を求めるために、より多くの有効桁を必要となる。**ex.1**, **ex.2** から、有効桁を上げさえすれば、真の gcd と近似 gcd の分離ができることがわかる。

■ f と df/dx の剰余列

ex.3-(a)

$$\begin{aligned} p_1 &= (x+1)(x-2) \\ &\quad \times (x-0.5)(x-0.501)(x-0.503) \\ p_2 &= dp_1/dx \end{aligned}$$

ex.3-(b)

$$\begin{aligned} p_1 &= (x+1)(x-2) \\ &\quad \times (x-0.5)(x-0.501)(x-0.501) \\ p_2 &= dp_1/dx \end{aligned}$$

この例は [1] に示されている悪条件剰余列で、剰余列の桁落ちが激しいのが特徴である。

gcd.3-(a)

$$\begin{aligned}
(7) \quad & 0.999999_6 e^{-6} x^4 - 2.00319_6 e^{-5} x^3 \\
& + 0.1548017_7 e^{-7} x^2 + 0.851198_6 e^{-6} x - 0.2764_6 e^{-6} \\
(13) \quad & 0.99999_5 e^{-5} x^3 - 1.5047_5 e^{-4} x^2 \\
& + 0.7548_5 e^{-5} x - 0.1262_5 e^{-5} \\
(22) \quad & 0.99999999_8 e^{-8} x^2 \\
& - 1.00266667_9 e^{-8} x + 0.25133433_8 e^{-8} \\
& \approx (x - 0.5)^2 \\
(32) \quad & 0.99999999_9 e^{-9} x - 0.50085714_9 e^{-9} \\
& \approx (x - 0.5) \\
(36) \quad & 3.719 - e_4^{e-9}
\end{aligned}$$

gcd.3-(b)

$$\begin{aligned}
(7) \quad & 0.999999_6 e^{-6} x^4 - 2.00159_6 e^{-5} x^3 \\
& + 0.1524005_7 e^{-7} x^2 + 0.850599_6 e^{-6} x - 0.2757_6 e^{-6} \\
(13) \quad & 0.99999_5 e^{-5} x^3 - 1.5023_5 e^{-4} x^2 \\
& + 0.7524_5 e^{-5} x - 0.12559_5 e^{-5} \\
(23) \quad & 0.99999999_9 e^{-9} x^2 \\
& - 1.00133333_{10} e^{-9} x + 0.250667_9 e^{-9} \\
& \approx (x - 0.5)^2 \\
(36) \quad & 0.99999999_{12} e^{-12} x - 0.50099999_{12} e^{-12} \\
& \approx (x - 0.5)
\end{aligned}$$

両例とも桁落ちが激しいが、安定に計算できている。

■ ex.4: 離れた根

ex.4-(a)

$$\begin{aligned}
p_1 &= (x+1)(x-2)(x-1.5)(x-0.500)(x-500) \\
p_2 &= (x-1)(x+2)(x+1.5)(x-0.501)(x-501)
\end{aligned}$$

ex.4-(b)

$$\begin{aligned}
p_1 &= (x+1)(x-2)(x-1.5)(x-0.500)(x-500) \\
p_2 &= (x-1)(x+2)(x+1.5)(x-0.500)(x-501)
\end{aligned}$$

ex.4-(c)

$$\begin{aligned}
p_1 &= (x+1)(x-2)(x-1.5)(x-0.500)(x-500) \\
p_2 &= (x-1)(x+2)(x+1.5)(x-0.501)(x-500)
\end{aligned}$$

ex.4-(d)

$$\begin{aligned}
p_1 &= (x+1)(x-2)(x-1.5)(x-0.500)(x-500) \\
p_2 &= (x-1)(x+2)(x+1.5)(x-0.500)(x-500)
\end{aligned}$$

これらはいずれも、 $x = 0.5$ と $x = 500$ の離れた点の辺りに根を持つ。

gcd.4-(a)

$$\begin{aligned}
(9) \quad & 0.9999_4 e^{-4} x^4 - 626.1_4 e^{-1} x^3 \\
& + 311.8_4 e^{-1} x^2 + 751.5_4 e^{-1} x - 375.8_4 e^{-1} \\
(15) \quad & 0.9999_4 e^{-4} x^3 - 0.5047_4 e^{-4} x^2 \\
& - 1.198_5 e^{-4} x + 0.6006_4 e^{-4} \\
(18) \quad & 0.99_2 e^{-2} x^2 - 0.46_2 e^{-2} x + 0.0_3 e^0 \\
(25) \quad & 0.999999_6 e^{-6} x - 0.499814_6 e^{-6} \\
& \approx (x - 0.5) \\
(36) \quad & 0.01495869683_{11} e^{-12}
\end{aligned}$$

gcd.4-(b)

$$\begin{aligned}
(9) \quad & 0.9999_4 e^{-4} x^4 - 626.1_4 e^{-1} x^3 \\
& + 311.4_4 e^{-1} x^2 + 751.4_4 e^{-1} x - 375.3_4 e^{-1} \\
(15) \quad & 0.9999_4 e^{-4} x^3 - 0.5041_4 e^{-4} x^2 \\
& - 1.1978_5 e^{-4} x + 0.5999_4 e^{-4} \\
(18) \quad & 0.99_2 e^{-2} x^2 - 0.46_2 e^{-2} x + 0.0_3 e^0 \\
(36) \quad & 1.0_{17} e^{-17} x - 0.5_{17} e^{-17} \\
& \approx (x - 0.5)
\end{aligned}$$

gcd.4-(c)

$$\begin{aligned}
(12) \quad & 0.99999999_7^{e-7} x^4 - 500.5005_7^{e-4} x^3 \\
& + 249.0998_7^{e-4} x^2 + 600.6707_7^{e-4} x - 300.36_7^{e-4} \\
(19) \quad & 0.999_3^{e-3} x^3 - 500.0_3^{e0} x^2 \\
& + 250.0_3^{e0} x - 1.029_4^{e-3} \\
(35) \quad & 0.99999999_9^{e-9} x^2 \\
& - 500.49982_9^{e-6} x + 249.910081_9^{e-6} \\
& \approx (x-500)(x-0.5) \\
(48) \quad & 0.99999999_9^{e-9} x - 499.999999_9^{e-6} \\
& \approx (x-500)
\end{aligned}$$

gcd.4-(d)

$$\begin{aligned}
(12) \quad & 0.99999999_7^{e-7} x^4 - 500.4999_7^{e-4} x^3 \\
& + 248.7999_7^{e-4} x^2 + 600.5999_7^{e-4} x - 299.9999_7^{e-4} \\
(18) \quad & 0.99_2^{e-2} x^3 - 500.0_2^{e1} x^2 + 240.0_2^{e1} x + 0.0_3^{e1} \\
(48) \quad & 1.0_{21}^{e-21} x^2 - 500.5_{21}^{e-18} x + 250.0_{21}^{e-18} \\
& \approx (x-500)(x-0.5)
\end{aligned}$$

ex.4-(a) と ex.4-(b) では $(x-0.5)$ だけが近似 gcd として得られ, $(x-500)$ の近似 gcd は得られていない. ex.4-(c) では $(x-500)(x-0.5)$ がまず得られ, 有効桁を上げると $(x-500)$ が近似 gcd として得られる. ex.4-(d) では $(x-500)(x-0.5)$ が近似 gcd として得られる.

■ ex.5: 離れた二つの近接重根

ex.5

$$\begin{aligned}
p_1 &= (x-0.500)(x-0.502)^2(x-500)(x-502)^2 \\
p_2 &= (x-0.501)(x-0.502)(x-0.503)(x-501)(x-502)(x-503)
\end{aligned}$$

$x=0.5$ と $x=500$ の付近に近接 3 重根を持つ例で, gcd として $(x-0.502)(x-502)$ を持つ.

gcd.5

$$\begin{aligned}
(10) \quad & 0.9999_4^{e-4} x^5 - 1005.4_5^{e-1} x^4 + 253760.0_5^{e1} x^3 - 506100.0_4^{e2} x^2 + 316760.0_5^{e1} x - 63500.0_4^{e1} \\
(20) \quad & 0.999999_6^{e-6} x^4 - 1003.007_7^{e-3} x^3 + 252007.9_7^{e-1} x^2 - 252132.9_7^{e-1} x + 63190.5_6^{e-1} \\
& \approx (x-500)^2(x-0.5)^2 \\
(29) \quad & 0.99999999_8^{e-8} x^3 - 503.0035_9^{e-6} x^2 + 504.009007_9^{e-6} x - 126.380133_9^{e-6} \\
& \approx (x-500)(x-0.5)^2 \\
(64) \quad & 1.0_{31}^{e-31} x^2 - 502.502_{31}^{e-28} x + 252.004_{31}^{e-28} \\
& \approx (x-502)(x-0.502)
\end{aligned}$$

22.3.2 有効剰余列と近似 gcd

前節の有効剰余列の計算から次のことがわかる.

- 剰余列の各要素が有効 gcd であるような有効桁が存在する.

●真の gcd は分離できる.

●真の gcd がない場合はもっとも近い近似根らしきものが有効 gcd として求まる.

しかし前節の有効 gcd を求めただけでは, それが近似 gcd であることの保証ができない. [1] に従い, 近似 gcd を,

$$p_1(x) = p_k(x)q_1(x) + r_1(x), \quad \deg(r_1(x)) < \deg(p_1(x))$$

$$p_2(x) = p_k(x)q_2(x) + r_2(x), \quad \deg(r_2(x)) < \deg(p_2(x))$$

を満たす $p_k(x)$ で, 近似の程度は $r_i(x)$ の係数の次数で判定する.

有効剰余列を用いると精度のいい剰余列が求められるので, この各要素 $p_k(x)$ を使って商と余りを求めてみるのが考えられる. ex-1.1 について有効桁 32 桁の有効剰余列は,

ex.1.1 の剰余列

$$(32) \quad p_6 = 0.999999999_{14}^{e-14} x - 0.50150157_{14}^{e-14}$$

$$(32) \quad p_5 = 1.0_{20}^{e-20} x^2 - 1.00163709_{21}^{e-20} x + 0.25081800_{20}^{e-20}$$

$$(32) \quad p_4 = 1.0_{24}^{e-24} x^3 - 1.00452278_{25}^{e-24} x^2 + 0.25639078_{24}^{e-24} x - 0.00206899_{22}^{e-24}$$

$$(32) \quad p_3 = 1.0_{29}^{e-29} x^4 - 1.00320068_{30}^{e-29} x^3 - 0.94867797_{29}^{e-29} x^2 + 1.20398129_{30}^{e-29} x - 0.30192256_{29}^{e-29}$$

この剰余列を用いて, 有効除算により商と余りを求める (p_1 についてのみ結果を示す) と

$$(32) \quad p_1/p_6 = 1.0_{13}^{e-12} x^4 - 3.00049842_{11}^{e-10} x^3 + 0.7512453_{8}^{e-8} x^2 + 3.25025_{7}^{e-6} x - 1.5014_{5}^{e-4} \\ \dots 0.0_{3}^{e-3}$$

$$(32) \quad p_1/p_5 = 1.0_{19}^{e-18} x^3 - 2.50036290_{16}^{e-15} x^2 - 0.49927424_{12}^{e-12} x + 3.00054442_{10}^{e-9} \\ \dots -8.164e - 4_{4}^{e-7} * (0.999_{3}^{e-3} x - 0.501_{3}^{e-3})$$

$$(32) \quad p_1/p_4 = 1.0_{23}^{e-22} x^2 - 2.49747721_{20}^{e-19} x - 0.50916355_{16}^{e-16} \\ \dots 3.00443273_{15}^{e-14} * (0.99999999_{14}^{e-14} x^2 - 1.00056239_{15}^{e-14} x + 0.25027904_{15}^{e-15})$$

$$(32) \quad p_1/p_3 = 1.0_{28}^{e-27} x - 2.49879931_{25}^{e-24} \\ \dots 0.69788079_{22}^{e-22} * (1.0_{21}^{e-21} x^3 - 1.00452278_{22}^{e-21} x^2 + 0.25639078_{21}^{e-21} x - 0.00206899_{20}^{e-22})$$

p_1/p_6 の有効剰余は 10^{-3} 程度の有効ゼロとなり, 10^{-3} 程度の近似 gcd であることが保証できる. さらに有効桁を上げて計算すれば, 有効ゼロの次数が下がり, この近似 gcd が真の gcd である高い保証が得られる.

p_1/p_5 の有効剰余は 10^{-3} 程度の有効数となり, さらに有効桁を上げて計算しても, 有効剰余の次数を小さくできない. p_5 は 10^{-3} 程度の近似 gcd であることが保証できる.

p_3 や p_4 については有効剰余の次数は入力多項式の係数の次数程度であるので, これは近似 gcd とは認められないことがわかる.

このように, 精度の高い有効剰余列を求め, 有効除算することで, 近似 gcd の判定が行なえること

わかる.

有効剰余算法では全ての近似 gcd が出てくるわけではない. しかし, 真の gcd を取り出すことができるので, 全体の算法は,

1. 真の gcd の分離
2. 最も近い近接 gcd の分離, 有効 gcd 計算を繰り返す.

となる. 近接 gcd の分離は [1] の提案にしたがって, Newton 法で行なえばいいと考える.

22.4 まとめ

本稿では, 不正確数の表現として有効数を提案した. その結果,

1. 計算誤差の混入を防ぐこと (逆に言えば誤差の入らない精度を求めること),
2. 演算結果の精度保証を行なうこと,
3. 計算精度の制御ができること,
4. 精度に関する構成ができること,

が実現できた. 2, 3 は多倍長の区間数でも実現できるが, 1, 4 ができることが, 本システムの特徴である.

有効数係数多項式算法の例として, 剰余列計算を行なった. 従来の算法をほとんど変えることなく, 精度のいい有効剰余列計算が実現できることを示した. 有効剰余列計算を近似 gcd 算法として見たとき, 近似 gcd の近似の程度を示すことができた. 今後, 各種算法について解析を行ないたい.

参考文献

- [1] T. Sasaki and M-T. Noda. Approximate Square-free Decomposition and Root-finding of Ill-Conditioned Algebraic Equation. *J. Inf. Proces.*, 12(2):159-168, 1989.
- [2] Shirayanagi, K., An Algorithm to Compute Floating Point Gröbner Bases, *Mathematical Computation with Maple V: Ideas and Applications* (Ed. T. Lee), Birkhäuser (1993), 95-106.
- [3] 白柳潔. 近似代数計算における安定化理論. 平成6年度 数式処理学会 (沼津), 1995.
- [4] K. Sirayanagi and M. Sweedler. A Theory of Stabilizing Algebraic Algorithms. Tech. Rept 95-28, Math. Sci. Inst., Cornell Univ., 1995.
- [5] M. Suzuki and T. Sasaki. A construction methods for ill-conditioned PRS. In *The Proc. of the 4th Riken symposium on Computer algebra and large-scale scientific computations*, pages 1-6, 1992.
- [6] 鈴木正幸. 有効精度付多項式と効率的な演算. 平成7年度 数式処理学会 (奈良), 1995.